

# Inhalt

1	Einleitung .....	5
2	Begriffe der objektorientierten Programmierung .....	7
2.1	Objekt, Klasse .....	7
2.2	Vererbung .....	8
2.3	Bindung, Polymorphie .....	9
3	Nicht-objektorientierte Erweiterungen in C++ .....	11
3.1	Kommentare .....	11
3.2	Typen .....	11
3.3	Der Typ bool .....	12
3.4	Deklarationen .....	13
3.5	Der Operator :: .....	13
3.6	Referenzen .....	14
3.7	Standard-Parameter .....	16
3.8	const .....	17
3.9	Inline-Funktionen .....	19
3.10	Überladen von Funktionen .....	19
3.11	Die Operatoren new und delete .....	20
4	Klassen .....	23
4.1	Begriffsbestimmungen .....	23
4.2	Syntax .....	23
4.3	Memberfunktionen .....	25
4.4	Konstante Memberfunktionen .....	26
4.5	Konstruktoren .....	27
4.6	Destruktoren .....	30
4.7	Initialisierung von Objektkomponenten .....	31
4.8	Konstruktor für komponentenweise Initialisierung .....	34
4.9	Objekte als Feldelemente .....	35
4.10	Befreundete Funktionen und Klassen .....	36
4.11	Überladen von Operatoren .....	38
4.12	Spezielle Operatoren .....	40
4.13	Die Operatoren new und delete .....	42
4.14	Der Zuweisungsoperator .....	43
4.15	Konversionsfunktionen .....	43
4.16	Explizite Konstruktoren .....	45
5	Iostream - Ein/Ausgabe .....	47
5.1	Ausgabe .....	49
	Manipulatoren .....	49
	Ausgabebreite und Ausrichtung .....	51
	Ganze Zahlen .....	51
	Gleitkommazahlen .....	51
	Logische Werte .....	52
	Weitere Manipulatoren .....	52
5.2	Eingabe .....	52
5.3	Unformatierte Ausgabe .....	53
5.4	Unformatierte Eingabe .....	53
5.5	Einstellung der Formatparameter .....	55
	Format-Flags .....	55
	Weitere Formatparameter .....	56
5.6	Statusabfrage .....	56
5.7	Benutzerdefinierte Erweiterungen .....	58

5.8	Dateibearbeitung	60
5.9	Umleiten von Strömen	63
5.10	Einschränkungen in den alten Iostreams	64
6	Statische Objektkomponenten	65
6.1	Datenkomponenten	65
6.2	Statische Memberfunktionen	67
7	Abgeleitete Klassen, Vererbung	69
7.1	Allgemeines	69
7.2	Deklaration abgeleiteter Klassen	69
7.3	Vererbung und Zugriffsrechte	71
	public-Ableitung	71
	protected-Ableitung	72
	private-Ableitung	73
7.4	Auflösung von Mehrdeutigkeiten	73
7.5	Initialisierung von Basisklassen	74
7.6	Multiple Vererbung und virtuelle Basisklassen	76
8	Virtuelle Funktionen	81
8.1	Frühe und späte Bindung	81
8.2	Deklaration	82
8.3	Abstrakte Basisklassen	86
9	Beispiele	89
9.1	Listen und Iteratoren	89
9.2	Objekte mit Namen	93
10	Zeiger auf Klassenkomponenten	97
11	Dynamische Typinformation	103
11.1	RTTI	103
11.2	Zugriff auf die Typinformation	103
11.3	Neue Casts	104
	static_cast	105
	dynamic_cast	105
	const_cast	106
	reinterpret_cast	107
12	Templates	109
12.1	Funktionstemplates	109
12.2	Klassentemplates	111
	Templates und friends	114
	Statische Komponenten	115
	Template-Memberfunktionen	115
	Eingeschachtelte Typdefinitionen	116
	Mehrdeutigkeiten	117
13	Namespaces	119
13.1	Zweck und Anwendung	119
13.2	Kumulation und Aliasbildung	121
13.3	Namespaces und Vererbung	121
14	Ausnahmebehandlung	123
14.1	Allgemeines	123
14.2	Explizite Deklaration	126
14.3	Spezielle Funktionen	127
15	Die Standard-Library	129
15.1	Header	129

---

15.2 Die Klasse string.....	129
Überblick .....	129
Ein- und Ausgabe.....	130
15.3 STL.....	131
Index .....	135

# Index

<b>A</b>		
abgeleitete Klasse	69	
Abgeleitete Klassen	69	
Ableitung	8	
anonyme union	12	
Aufrufoperator	41	
<b>B</b>		
bad()	57	
bad_cast	106	
badbit	56	
Basisklasse	69	
Bindung	81	
bool	12	
boolalpha	52	
Botschaft	7,98	
<b>C</b>		
Cast	11,43	
catch	123,124,125	
cerr	47	
cin	47,52	
class	23	
clear()	57	
clog	47	
clone	85	
close	62	
const	17,26	
Memberfunktionen	26	
const_cast	105,106	
copy constructor	35	
cout	47	
<b>D</b>		
Dateibearbeitung	60	
Default-Konstruktor	28	
Deklarationen	13	
for	13	
if	13	
switch	13	
delete	20	
Destruktor	30	
expliziter Aufruf	31	
virtueller	85	
dynamic_cast	104,105	
dynamischer Typ	105	
<b>E</b>		
enum	65	
eof()	57	
eofbit	56,63	
Exceptions	123	
explicit	46	
<b>F</b>		
fail()	57	
failbit	56	
Fehlerausgabe	47	
Felder von Klassenobjekten	35	
fmtflags	55	
Freundschaft	71	
friend	36	
Funktionstemplates	109	
<b>G</b>		
gcount	54	
get	53,130	
getline	54	
globaler Namensraum	120	
good()	57	
<b>H</b>		
Header-Dateien	129	
<b>I</b>		
ifstream	60	
ignore	54	
Indexoperator	40	
information hiding	7	
Initialisierung	32,35,74	
komponentenweise	34	
Reihenfolge	75	
virtuelle Basisklasse	77	
Initialisierungsliste	32,74	
Reihenfolge	32	
Initialisierungsphase	33	
Inkrement- und Dekrementoperatoren	40	
Inkrement-/Dekrementoperatoren	40	
inline	19,25	
ios		
app	61	
ate	61	
beg	62	
binary	61	
cur	62	
end	62	
in	61	
out	61	
seek_dir	62	
trunc	61	
iostreams	47	
Ausgabe	49	
Benutzerdefinierte Erweiterungen	58	
Eingabe	52	
istream	47,52	
istreamstringstream	130,131	
Iterator	91,131	
<b>K</b>		
Kapselung	7	
Kapselungsprinzip	69	

Klasse	7,23	peek	55
Klassen		POD	24
polymorphe	106	Polymorphie	9
Klassentemplates	111	Positionierung	62
Konstruktor	27	private	23
Konversion	43	private Ableitung	71
Konversionsfunktionen	43	private Klasse	38
<b>L</b>		private Komponenten	23
late binding	9	protected	71
list<>	132	Ableitung	72
Log-Ausgabe	47	public	23,69
logische Konstantheit	27	put	53
<b>M</b>		putback	55
Manipulator	49	<b>R</b>	
benutzerdefiniert	59	rdstate	57
dec	50	read	54
endl	51	read-only	26
flush	51	Referenz	14
hex	50	rein virtuelle Funktion	86
oct	50	reinterpret_cast	105,107
parameterlos	50	RTTI	105
parametrisiert	50,60	<b>S</b>	
setfill	51	Schnittstelle	86
setprecision	51	Scope-Operator	24
setw	51	seekg	62
map<>	132	seekp	62
Mehrdeutigkeiten	73	set_terminate	127
Memberfunktionen	23,25	set_unexpected	127
Methoden	7,23	Shift-Ketten	49
Methodenprotokoll	7	Shiftoperatoren	40
multiple Vererbung	8	Signatur	19
<b>N</b>		späte Bindung	9
Namenskollisionen	73	Speicher-Leck	30
namespace	119	Standardausgabe	47
std	129	Standardeingabe	47
und Vererbung	121	Standardformat	49
new	20,42	static_cast	104
<b>O</b>		Statische Memberfunktionen	67
Objekt	7,23	statische Memberfunktionen	98
ofstream	60	Statische Objektkomponenten	65
open	61	Statusabfrage	56
Operator		Statuswort	57
<<	49	STL	131
=	43	streamoff	62
>>	52	streampos	62
delete	42	string	129
new	42	struct	24
Operatorfunktionen	38	<b>T</b>	
Operatorschreibweise	39	tellg	62
ostream	47,49	tellp	62
ostreamstream	130	Templatefunktion	109
<b>P</b>		Templateklassen	111
Parameterübergabe per Referenz	14	Template-Memberfunktionen	115
		Templates	

---

statische Komponenten. ....	115
und friends . . . . .	114
temporäre Variable . . . . .	15
terminate . . . . .	127
this . . . . .	25,67,97
throw . . . . .	123,126
try . . . . .	123
Typkonvertierungen . . . . .	11
<b>U</b>	
Überladen . . . . .	19,74
Operatoren . . . . .	38
unexpected . . . . .	127
Unformatierte Ausgabe. . . . .	53
Unformatierte Eingabe . . . . .	53
using . . . . .	119,121
<b>V</b>	
vector<> . . . . .	132
Vererbung . . . . .	8,69
multiple. . . . .	69
virtual . . . . .	82
Virtuelle Basisklassen. . . . .	76
virtuelle Funktion . . . . .	83
Virtuelle Funktionen . . . . .	81
<b>W</b>	
wide character . . . . .	47
write . . . . .	53
<b>Z</b>	
Zeiger auf Klassenkomponenten. . . . .	97
Zeigeroperator. . . . .	41
Zugriffsrechte . . . . .	71
Zuweisungsoperator . . . . .	43
Zuweisungsphase . . . . .	33